

PAPER  
NETWORK PROXIMITY FOR CONTENT DISCOVERY

## Network Proximity for Content Discovery

<http://dx.doi.org/10.3991/ijim.v9i4.4657>Yousef Ibrahim Daradkeh<sup>1</sup> and Dmitry Namiot<sup>2</sup><sup>1</sup>Prince Sattam bin Abdulaziz University, KSA<sup>2</sup>Lomonosov Moscow State University, Moscow, Russia

**Abstract**—The paper describes our approach for using wireless sensors on mobile phones for delivering new data to mobile subscribers. We propose a new practical approach for social context-aware data retrieval based on mobile phones as a sensor concept. This approach uses Wi-Fi and Bluetooth modules located on mobile phones as sensors for getting proximity information that can open (discover) access to any user generated content or content published in the social networks. A special mobile service (context-aware browser client for Android) can present that information to mobile subscribers. The potential use-cases for the proposed approach include all projects associated with hyper-local news data. For example, news services in Smart City projects, proximity marketing, indoor data delivery, etc..

**Index Terms**—Context-aware computing; Bluetooth; Wi-Fi; proximity; social networks; data mining.

## I. INTRODUCTION

**Context awareness.** The first paper, that introduced the term ‘context-aware’, refers to context as location, identities of nearby people and objects, and changes to those objects [Schilit et al., 1994]. As per the more generic definition [Day, 2001], context is any information that characterizes the situation of an entity. An entity in this definition could be a person, place, or object. Objects are considered to be relevant to the interaction between a user and an application. This definition is more practically oriented, and it lets simply enumerate the context during development phases.

Context-related information consists of some user identifications (e.g., in the simplest case - user profile and preferences), their current and past location info. Context-related data can include information about user’s mobile devices (e.g., current and past state), objects and processes in the user’s proximity. Context-related information can cover behavioral history too.

But in any case, context awareness is a complementary element to location awareness. Location serves as a determinant characteristics, context introduces the ability to work with any moving entities. Context awareness is as a term from pervasive (ubiquitous) computing and describes the processing for linking changes in the environment.

**Smart phones as sensor concept.** The proliferation of smart phones throughout has provided an increasingly ubiquitous platform for intelligent services. We mean a growing popularity of applications that use mobile phones for performing local data acquisition and aggregation functions. In our article we are also using mobile phones (smart phones) as sensors. Indeed, mobile phones are probably the most obvious candidates for the mass sensing products. In various research papers smart phones become a useful tool in the social sciences (sociology and psy-

chology), urban studies (city sense applications), media studies and technology assessment. There are several attempts to create standards for data acquisition. For example, Sensor ML [Trossen et al., 2005] can be used for providing general information about sensors and sensor groups. This tool enables discovery of sensors as well as processing and analysis of sensor measurements. It supports detailed information about sensors, like identification, description, location, constraints, measurement characteristics, etc.

The modern smart phones support different build-in sensors. In our SpotEx (Spot Expert) approach we choose only one particular kind of sensors - wireless network sensors. Yes, wireless modules in the modern smart-phones are data transmitters. But we can use them as sensors. It is the key moment. We do not use data transmitting features. We use the fact that Wi-Fi access point (Bluetooth node) is visible. Two mobile users (two mobile phones) capable to see the same Wi-Fi access point (Bluetooth node) are in the proximity. Just due to the limited distance for Wi-Fi access point (Bluetooth node in discovery mode) announce. It is where our novelty is. And it is a very practical solution that lets us work with the most widely adopted sensors. All the modern smart phones support wireless networks (read - Wi-Fi and Bluetooth).

**Network proximity as a key notion.** What kind of information can we get from the wireless sensors? For example, Wi-Fi based indoor positioning systems can provide location information. But this approach usually requires the calibration for the environment. Our choice is network proximity [Namiot et al., 2012]. According to our selection of Wi-Fi sensors, this proximity could be described via the data measured by sensors. For example, the visibility for Wi-Fi access points, signal strength, etc. The same is true for Bluetooth nodes in so-called discovery mode. So, more precisely, we can describe our task as using network proximity info based on the wireless sensors in mobile phone for getting (discovering) new content for mobile users. It could be either some user generated content linked to sensors info or social streams. The change in the signal strength lets us associate some data with the values. That is the main difference from indoor positioning based on the Wi-Fi. In our approach we do not need the position info. User-defined data chunks will be linked to the changes detected in the network environment.

The novelty of this approach is based on the way the proximity is defined. For existing projects the proximity is always defined via geo-coordinates. Even for indoor positioning based on Wi-Fi, the proximity is based on the geo-coordinates calculations. Wi-Fi based indoor positioning uses the preliminary scene preparation (so-called radio map). This map can be used later for geo-calculation (as-

suming we know the geo-coordinates for access points). So, regardless of the sensors (GPS or Wi-Fi) the final calculation is always geo-related. The preliminary scene preparation (radio map) is, probably, the biggest weaknesses. It takes a time, it is expensive and maps should be constantly updated. With our approach, we suggest to compare wireless fingerprints only. In our model (SpotEx) data are directly linked to wireless fingerprints. If any mobile phone (mobile user) can see the predefined Wi-Fi access point (Wi-Fi module in the phone accepts broadcasted Wi-Fi announce), than two nodes are in the proximity, regardless of their geo-coordinates. The latest development from Apple (iBeacons) has got the similar conception. But Apple uses some special wireless tags. Our model uses ordinary smartphones and existing wireless nodes.

The rest of the paper is organized as follows. Section 2 contains an analysis of existing projects that combine sensors and user generated data. On the first hand, we consider social streams. Section 3 describes some use cases. Section 4 describes SpotEx approach used as a conceptual and technological base for linking sensors and user generated content. We present in this section the main contribution of this paper – procedure-based language (library of predicates) for describing network proximity based content discovery. Section 5 discusses the content in proximity based systems. We provide here the finalized version of content description for network proximity model.

## II. ON SENSORS AND USER GENERATED DATA

Our early papers about SpotEx (e.g., [Namiot et al., 2012]) describe data integration for user-generated texts (HTML chunks) and wireless sensors. We will describe it more detailed below. Many other researches devoted to sensors and social networks integration. There are several drivers for this process.

Firstly, allow the actors in the social network to publish their data and subscribe to each other's data either directly, or indirectly after discovery of useful information from such data [Miluzzo et al., 2008]. Obviously, that such collaborative sharing on social networks can increase real-time awareness of different users about one another. It provides also the basic information for analysis and understanding the global behavior of different actors in the social networks.

Integrating sensors and social networks could lead to better understand the aggregate behavior of self-selected communities or the external environment in which these communities function. Sensors can introduce measurements (metrics) for this process.

For example, CenseMe combines the inference of the presence of individuals using sensor-enabled mobile phones with sharing of information through social networking applications such as Facebook [Miluzzo et al., 2008]. This application is mostly oriented to process of sharing information from sensors. Sensors data processing may include traffic and environmental pollution levels information. We should mention in this context the original development in Reality Mining [Madan et al., 2010]. It used Bluetooth transceivers from mobile phones for linking user's activity to the places.

The typical example for sensing data processing is City Sense application [Murty et al., 2008]. It collects data

extracted from GPS-enabled cell phones and taxi cabs with idea to determine where the people are, and then delivers this information to subscribers with mobile devices. This application is designed to track important trends in the behavior of people in the city.

The City Sense application provides a social networking version also. It is a tool based on a collaborative filtering. The application stores a user's personal history. This personal history could be used for predicting where other similar users might be. In other words, it can provide recommendations for users about possible places to visit based on their past behavior.

On the social side, this application introduces own social network. In the context of this article, we think it is a big disadvantage. In our research, we are more interested in the integrating sensing info with existing social networks. For example, using sensing information with Twitter streams or Facebook data feeds.

Usually, merging data from sensors and social streams mean simply adding (posting) data from sensors to the special streams. See, for example, CenceMe above. Our idea is to return to the main point of view – sensors are forming a new dimension of the existing data. In our particular case (social data streams), all information we can get from sensors should provide a new dimension to the posts in the social streams. This new dimension could be used for the data analysis (for the classification at the first hand).

The abstract model often used for social data is so called spatio-temporal thematic stream. In practice, such stream could be defined as an uninterrupted flow of data points, each point of which can explicitly be assigned a spatio-temporal coordinate, and an application theme.

For merging (pairing) sensors data and social streams (e.g., Twitter's stream), we can present thematic stream as data (topics) discovered from the tweets and data recorded from our sensors. In our case the set of sensors is limited (wireless sensors only), so our possible data (data types) could be easily specified.

On the abstract level we have the three basic types of attributes (so called where-when-what scheme): location, time and topic [Namiot et al., 2013a]. It is how our spatio-temporal points could be described. The stream is a collection of different spatio-temporal thematic points. Each point has got three the above mentioned attributes: location, time stamp and topic. The topic here is some content, extracted from the tweets. For example, it could be simply a hash tag for Twitter's stream.

$$\text{Point} = \{ \text{location info, time stamp, topic} \}$$

$$\text{Stream} = [ \text{Point}_1, \text{Point}_2, \dots ]$$

Usually, the location is simply a pair of latitude, longitude. With our network proximity model, location is replaced with proximity. We mean here the relative position regarding the network nodes. E.g., for Wi-Fi networks it could be a proximity to some Wi-Fi access point (points). So, in our definition:

$$\text{Point} = \{ \text{proximity info, time stamp, topic} \}$$

Such definition lets us link topics (read – text extracted from the posting in the social networks) and wireless

nodes. The most often used tasks we can deploy these data are classification and aggregation. What are hot topics around the selected access point? Who are nearby writers? And so on.

Two main points should be mentioned here:

a) For social streams proximity provides more granulated (more precise) location information than the standard geo-coding (especially for indoor)

b) This approach is not necessarily linked to the static wireless nodes only. It is not about the statically fixed Wi-Fi router, for example. Wi-Fi node used for proximity calculation could be opened right on the mobile phone. Most of the modern smartphones support Wi-Fi hot spots configuring.

c) This approach does not touch the connectivity. For examples, access point in the mobile phone could be password protected and actually disallows public connections. But broadcasted (visible) SSID is enough for our model.

The same is true for Bluetooth. There is even one practical advantage – we can open Bluetooth node on the mobile phone programmatically (Figure 1). Note again, just because this approach does not touch the connectivity, opening Bluetooth node does not compromise the device. It is yet another form for setting own “location” info. So, Bluetooth node (Wi-Fi access point) informs the nearby users about owner’s presence. It could be very close to the “normal” check-in (a record in the social network, which points to some location) if we associate our data chunk with some social network info. E.g., data chunk, associated with Bluetooth node could be simply a link to Facebook (LinkedIn) profile. In this case, opened network node lets other mobile users in the proximity gets presence information about visitors (e.g., on some conference).

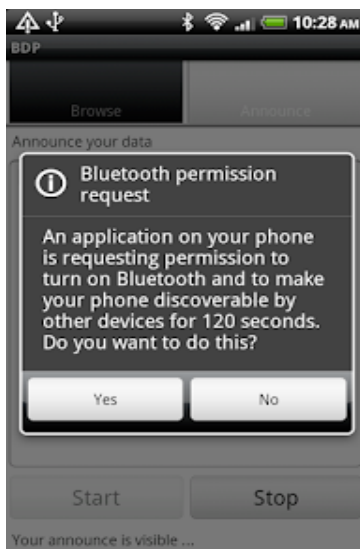


Figure 1. Bluetooth node on the phone (Android).

### III. IN-PROXIMITY CHECK-IN

The process of linking sensors and data from social network is described in our papers [Namiot, et al., 2013a, Namiot et al., 2013b]. Our idea for introducing proximity to social stream is based on two things: a special definition for places and a special form of linking social streams and places. In other words, it is based on the special check-ins.

Traditionally, for “places” in the social networks the key characteristic is geo location (latitude and longitude pair). It could be already problematic for indoor applications, when many places within a building could be in the same geo position. Our idea is to remove places from the social networks and make them completely dynamic. Certainly, it should work as an additional element for the “traditional” places. We will describe our “new places” separately and define them via proximity (network proximity in our case) attributes rather than via geo coordinates.

Proximity based definition for places means that each place should be defined via some metric, introduced to our network. As a base for metric for Wi-Fi proximity we can use our snapshot for wireless environment. It could provide:

1) counter, which shows how many networks are visible at this point;

2) vector with obtained measurements of the each network e.g., SSID, RSSI (signal level), etc.

A typical check-in record in social network is some message (post, status) linked to the particular location (place). In other words, it is some specialized message. For example, it could be just a geo-tagged post in the simplest case. What are the reasons for members in social networks to use such special kind of messages? In the most cases, it is stimulated by the business. Practically, with check-in user simply posts advertising for the business in exchange for some benefits. Sometimes it could be used by users themselves for social connection. Check-ins let other members of social graph knows where I am and see where my friends are.

Check-in could be customized. A business entity can create own forms for check-in records [Namiot et al., 2013a], but even the special check-ins are always part of the social stream.

Our idea is to create a new type of check-in records. Because our “places” are separated from the social network, our “new check-ins” will be separated from the social stream too. It means that we provide a separate database that just contains a list of accounts from the social network being checked-in (read – concentrated) at any particular moment nearby some place. It is a temporal database, check-in records could be changed constantly. And this database does not contain the social stream itself. It contains public links to social profiles only.

Now, suppose that we have a mobile application that lets users confirm their social network identity and link that identifies with wireless network info. This application fills our external temporal database, so it contains social network identity confirmation and appropriate network info. This pair (confirmed identity and network info) is simply a new form of check-in. This new check-in is an “external” entity in the social network. Our application does not post data back to the social network. It keeps check-in data outside the social stream (in the own database). So, in the terms of privacy this check-in does not affect (does not touch, actually) account settings in the social network.

What are the reasons for users to participate in new check-ins? Actually, they are the same as for “old” traditional check-ins. Sometimes it could be stimulated by the business. A business entity can use that information for statistics and deliver some benefits in exchange for

“check-in”. Certainly, it could be used for the social connections - it lets other members know where the user is and see who else is here. It is illustrated in Figure 2.

What can we do with this external database of check-ins? On the first hand, we can provide a list of other people at any particular location. Actually, it is always a list of people at “this” location only. Note again, that our proximity based system does not provide a traditional list of places. Each our “place” is a snapshot of Wi-Fi (Bluetooth) environment (visible access points and RSSI). Obviously, all the attributes are dynamical in this definition. So, any user can see only check-ins at the same place he is checked in himself. In this approach any user is simply unaware about other “places” unless he moves and makes a new check-in.

At the second – we can show (search) social streams nearby. Via public API we can read data feed for users (if it is possible, of course, and an appropriate stream is not protected). This system can keep the full respect to the existing privacy settings in social networks. And it is a typical content discovery process based on social streams and network proximity.

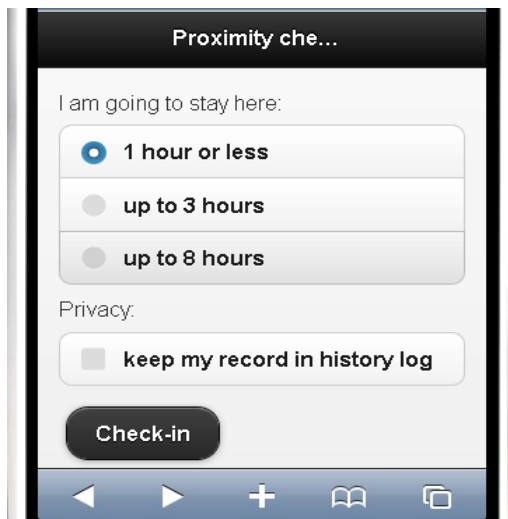


Figure 2. In-proximity check-in (Namiot et al., 2011a).

As a practical example, we can mention tweeting from the conferences. The common approach nowadays is to publish statuses (messages) with some hash tag. With sensor-based check-ins we can simply show public posts from the nearby people without the need to introduce a special markup (like hash tags) for messages. Note, that in this case messages in the social streams do not need geo-tags too.

#### IV. SPOTEX AS A NETWORK PROXIMITY CONTENT DISCOVERY

**What is SpotEx?** Originally, the main idea of SpotEx service (Spot Expert developed by Dmitry Namiot) was described by the authors in an article published in NGMAST-2011 proceedings [Namiot 2011b]. Originally, SpotEx started as a new approach for indoor navigation and migrated to the context-ware data discovery after that. This section describes the latest development in SpotEx.

SpotEx model is based on the ideas of network proximity. Wi-Fi hot spots (Bluetooth nodes) work in this model as presence sensors. The service supports a data-

base of production rules (if-then clauses) associated with Wi-Fi hot spots and Bluetooth nodes.

Technically, the same schema works transparently for both Wi-Fi and Bluetooth nodes. The only difference measured in our experiments is discovery time. Mobile phone (Android, for example) detects Wi-Fi nodes several times faster than Bluetooth nodes. Figure 3 illustrates this approach for wireless node right on the phone.

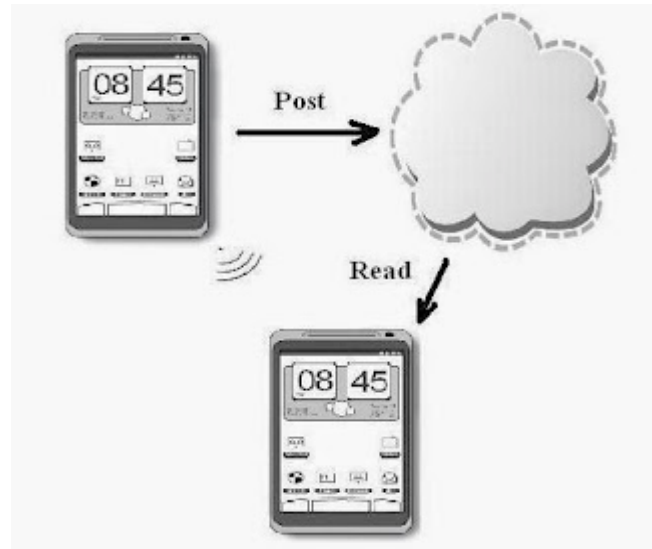


Figure 3. In-proximity data access

**On a practical execution.** We can use any existing wireless network (or networks, especially created for this service – the most interesting case, see remarks above about mobile hot spots) and add some rules (messages) to that network. More precisely, we add rules depend on the visible network environment. Any rule’s message (conclusion) is just some text that should be delivered (discovered, opened) to the end-user’s mobile terminal as soon as an appropriate rule is fired (e.g., network’s SSID, for example, is getting detected via our mobile application). In other words, our collection of rules is a set of operators like:

*IF IS\_VISIBLE(‘myshop’) AND TIME\_WITHIN(2pm, 3pm) THEN {present the coupon info}.*

This rule states that if Wi-Fi network with SSID *myshop* is visible and time of the day is from 2pm to 3pm, then the application should show to mobile user HTML code from the rule’s body. Alternatively - code block (data snippet) *{present the coupon info}* contains HTML code delivered to mobile browser for mobile subscribers who can see the access point *myshop* within 2pm-3pm time frame.

The word “delivered” here is a synonym for “available for reading/downloading”. For the end-users the whole process looks like an automatic (and anonymous) check-in. And the key moment for the future development is the fact that the place for check-in is defined via network elements (wireless nodes – see Section 2).

How our productions data store (base of rules) looks like? Each rule looks like a production (if-then operator). The conditional part is a logical combination (AND, OR, NOT) of basic predicates. Each predicate (*boolean func-*

tion) describes one aspect of proximity. Let us present the classes for predicates. Each class of predicates on the programming level will be built as a set of Boolean functions. So, finally, we will provide a procedure-based language for describing network proximity.

At his moment we can list the following relations:

Attributes of network nodes  
Date/Time  
Client's identification  
History of scanning  
Behavior

Note, that the whole schema in SpotEx is completely anonymous. It looks like a browser and does not require from mobile users to be authenticated. In other words, it is not a social network. So, our context-aware browser has no information about the user. But in the same time, MAC-address for mobile client is known. It is a direct parallel with web browsing, where the visitor's IP address is known.

The choice for attributes of network node is simple. We have SSID for Wi-Fi access point, name for Bluetooth point, MAC-address and signal strength (RSSI). So, we can suggest the following functions:

IS\_VISIBLE (name)  
IS\_VISIBLE (name, MAC-address)  
CLOSER\_THAN (name1, name2)  
CLOSER\_THAN (name1, MAC1, name2, MAC2)  
SIGNAL\_RANGE(name, R1, R2)  
SIGNAL\_RANGE(name, MAC-address, R1, R2)

Function IS\_VISIBLE () returns a true value if the network (point), described by the name or by the name and MAC-address is visible. Otherwise, it returns false.

Function CLOSER\_THAN() compares the signal strength for two Wi-Fi access points (Bluetooth nodes) and returns true if RSSI for the first element is bigger than second.

SIGNAL\_RANGE() function returns true if signal strength for the given Wi-Fi access point (Bluetooth node) is within the given interval.

Logical functions for data and time comparisons are obvious.

IS\_MON() IS\_TUE() IS\_WED() IS\_THU() IS\_FRI()  
IS\_SAT() IS\_SUN() TIME\_WITHIN(t1,t2)

Functions from IS\_MON() till IS\_SUN() check day of the week. Function TIME\_WITHIN() returns true if the current time is within the given interval (hours and minutes). We should note also that rules are checking on the client side (in mobile phone). So, time here is a time for mobile OS. SpotEx does not presume any time synchronization.

Because any fingerprint for wireless device describes some place (see section 2), we can discover repeated visits and frequent visitors. "Visit" for wireless node is a fact, that this node was visible during some scanning in SpotEx. As soon as some wireless access point is visible dur-

ing the scanning process, SpotEx records this information in log file. This log file is a direct analogue for web log [Suneeha 2009]. Obviously, it is important information for retail, for example. So, history-based predicates should help create rules for getting content depending on historical performance. Functions are:

FIRST\_VISIT(days)  
COUNT\_VISITS(days,n1,n2)

Function FIRST\_VISIT() returns *true* if it is a first visit during the given value for last recorded days. Argument's value 0 corresponds with the first visit in history. Function COUNT\_VISITS() checks that count of visits is within the given interval (n1, n2) for the last days. Visits will be checked against the current fingerprint.

Another aspect of the history of visits is that the administrator (business) located in a particular place can cause the issuance of a content not only by how often the mobile subscriber visiting this particular place, but also by how often he visited (or not visited) other places. So, we can add the following set of functions:

VISITED(name)  
VISITED(name, MAC)  
COUNT\_VISITS(name,days,n1,n2)  
COUNT\_VISITS(name,MAC,days,n1,n2)

Function VISITED() returns true, if visitor sometime in his history, visited place described by the given name or name and MAC-address. COUNT\_VISITS() checks that count of visits for the given wireless point is within the given interval (n1, n2) for the last days.

The list of basic functions is not fixed finally, of course. It corresponds with the current state of development.

Note also, that this model is different from Bluetooth Low Energy based solution, proposed by Apple as iBeacon. The main difference is the fact that iBeacons tags are available for the one application only (each application should be configured for some set of tags). With SpotEx each network node plays a role of a tag. And it is up to application (mobile users) to decide which tags (associated data) are interested for any particular task.

## V. CONTENT IN PROXIMITY-BASED APPLICATIONS

As it is mentioned above, our context-related info is a set of rules. The left part (condition) of the each production (rule) is explained above and presents some logical expression. Let us see what we can present in the right part (in the conclusion).

Code block (data snippet) *{present the coupon info}* contains HTML code delivered to the mobile browser.

Each such snippet has got a title (text) and some HTML content (it could be simply a link to an external web site for example). Snippets present coupons/discount info for malls, news data for campuses, etc. Technically, any snippet could be presented as a link to any external web site/mobile portal or as a mobile web page created automatically by the rule editor from SpotEx.

For the data presented as links to external existing mobile sites (portals), SpotEx works as some universal discovery tool. De facto, it lets mobile subscribers to be



aware about context-relevant web resources. Owners of the web resources can describe own mobile web sites via network nodes related rules rather than present for them individual QR-codes or NFC-tags, for example [Rouillard 2008].

For describing some content directly in SpotEx, the whole system works in this part as a content management system (CMS). For each existing data snippet SpotEx rule editor creates automatically mobile web page. SpotEx middleware hosts that page on the own server.

The SpotEx mobile application creates on the fly dynamic HTML page from titles (according to rules that are relevant in the given context). It is what mobile users will see. Technically, it is a classic rule based expert system. It matches existing rules against the existing context and makes the conclusions. Existing content here presents a snapshot for Wi-Fi environment. It is a list of hot spots with attributes. And conclusion here is a mobile web page assembled from the individual titles.

For mobile users the whole process looks like an ordinary browsing. The only difference is browser's awareness about hyper-local content. It is a typical example of context-aware data retrieval [Brown et al., 2001].

Besides the ordinary mobile sites as data snippets, our system can suggests some set of predefined mobile templates. For example:

- link to the profile in social network
- link to the data feed in social network
- vCard form for contact information sharing

It is illustrated in Figure 4.

Link to profile lets share Facebook (Twitter) pages for those who are in the physical proximity, links to social data feed helps share Twitter feeds on the conference, for example. Actually, all this is again about the discovery. How to share own contact data with people in the physical proximity? It is the typical question that could be answered with SpotEx.

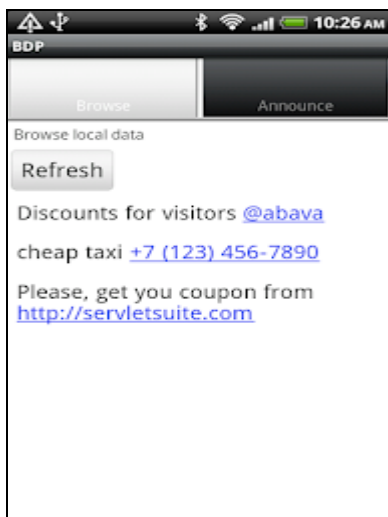


Figure 4. In-proximity browser

The possible use cases currently are circled around hyper-local news data. In proximity marketing one shop can deliver deals/discount/coupons right to mobile terminals as soon as the user is near some predefined access point.

We can describe this feature as “automatic check-in” for example. With traditional systems (Foursquare, Facebook Places, etc.) users set current place manually or via API in order to get deals info. With SpotEx mobile subscriber can collect deals automatically. For example, hyper local news in SmartCity projects could be linked to the public available networks and being delivered via that channel, etc.

Especially, we would like to point attention to the most interesting (by our opinion, of course) uses case: Wi-Fi hot spot being opened right on the mobile phone. Most of the modern smart phones let users open Wi-Fi hot spots. We can associate our rules to such hot spot (hot spots) and so our messages (data snippets) become linked to the phones. It is a practical example of the dynamic location based system [Jose et al., 2003]. Services follow to the moved phone.

This uses case is the most transparent demonstration of SpotEx model. It shows that ordinary smart phone is enough for creating a new dynamic information channel. There is no infrastructure except the smart phone itself.

## VI. CONCLUSION

This paper describes network proximity based model in context-aware data discovery for mobile users. This model is based on the ideas of network proximity and introduces a new form of data discovery for both user generated data and social streams. The service can use existing as well as the specially created network nodes as presence triggers in the process of discovering the relevant content for mobile subscribers. The proposed approach is completely software based and offers a practical and easy to use implementation of context-aware data retrieval for mobile subscribers.

The proposed service could be used for data discovery and distribution hyper-local news data in Smart City projects, for providing commercial information (proximity marketing - deals, discounts, coupons) in malls, for smart indoor navigation, etc.

As a main contribution of this paper, we provide here the description of procedure-based language (library of predicates) that could be used as a generic tool for network proximity based content discovery. As the second contribution, we complete here the description of procedures for content discovery in network proximity based applications.

## ACKNOWLEDGMENT

We Authors would like to thank Deanship of Scientific, Prince Sattam Bin Abdul Aziz University, Saudi Arabia for providing financial assistance to do this research a successful manner.

## REFERENCES

- [1] G. Schilit and B. Theimer (1994) “Disseminating Active Map Information to Mobile Hosts”. IEEE Network, 8(5) pp. 22-32 <http://dx.doi.org/10.1109/65.313011>
- [2] A. Day, (2001) "Understanding and Using Context" Human-Computer Interaction Institute. Paper 34. <http://repository.cmu.edu/hcii/34>
- [3] C. Bolchini1, G. Orsi, E. Quintarelli, F. A. Schreiber, and L. Tanca (2011) Context modeling and context awareness: steps forward in the context-addict project, IEEE Data Eng. Bull., 34(2): pp. 47–54

- [4] D. Trossen and D. Pavel (2005) Building a ubiquitous platform for remote sensing using smartphones Mobile and Ubiquitous Systems: Networking and Services, pp. 485-489
- [5] D. Namiot and M. Sneps-Sneppe (2012) "Proximity as a service". In Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on pp. 199-205. IEEE. <http://dx.doi.org/10.1109/BCFIC.2012.6217947>
- [6] iBeacon experiments (2014) <http://blog.nerdery.com/2013/11/nerdery-labs-ibeacon-experiments/> Received: Jun, 2014.
- [7] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell (2008) Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application, SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems, pp. 337-350 <http://dx.doi.org/10.1145/1460412.1460445>
- [8] A. Madan, S. Moturu, D. Lazer, and A. Pentland (2010) Social sensing: obesity, unhealthy eating and exercise in face-to-face networks, WH '10 Wireless Health pp. 104-110.
- [9] Murty, R.N., Mainland, G., Rose, I., Chowdhury, A.R., Gosain, A., Bers, J., and Welsh, M. (2008) CitySense: An Urban-Scale Wireless Sensor Network and Testbed Technologies for Homeland Security, IEEE Conference on, pp. 583 – 588.
- [10] Namiot, D., & Sneps-Sneppe, M. (2013a). Social streams based on network proximity. International Journal of Space-Based and Situated Computing, 3(4), 234-242. <http://dx.doi.org/10.1504/IJSSC.2013.058375>
- [11] Namiot, D., & Sneps-Sneppe, M. (2013b). Wireless Networks Sensors and Social Streams. In Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on (pp. 413-418). IEEE. <http://dx.doi.org/10.1109/waina.2013.27>
- [12] D. Namiot and M. Sneps-Sneppe (2011a) "Customized check-in Procedures", Smart Spaces and Next Generation Wired/Wireless Networking Lecture Notes in Computer Science, 2011, Volume 6869/2011, pp. 160-164 [http://dx.doi.org/10.1007/978-3-642-22875-9\\_14](http://dx.doi.org/10.1007/978-3-642-22875-9_14)
- [13] D. Namiot and M. Schneps-Schneppe (2011b) "About location-aware mobile messages", Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011, pp. 48-53 <http://dx.doi.org/10.1109/NGMAST.2011.19>
- [14] Suneetha, K. R., & Krishnamoorthi, R. (2009). Identifying user behavior by analyzing web server access log file. International Journal of Computer Science and Network Security, 9(4), 327-332.
- [15] Pei, Jian, et al. (2000) "Mining access patterns efficiently from web logs." Knowledge Discovery and Data Mining. Current Issues and New Applications. Springer Berlin Heidelberg, pp.396-407. [http://dx.doi.org/10.1007/3-540-45571-X\\_47](http://dx.doi.org/10.1007/3-540-45571-X_47)
- [16] X. Yu, M. Li, D. Gyu Lee, K. Deuk Kim, and K. Ho Yu (2012) Application of Closed Gap-Constrained Sequential Pattern Mining in Web Log Data, ADVANCES IN CONTROL AND COMMUNICATION Lecture Notes in Electrical Engineering, Volume 137, pp. 649-656, [http://dx.doi.org/10.1007/978-3-642-26007-0\\_80](http://dx.doi.org/10.1007/978-3-642-26007-0_80)
- [17] Rouillard, J. (2008). Contextual QR codes. In Computing in the Global Information Technology, 2008. ICCGI'08. The Third International Multi-Conference on (pp. 50-55). IEEE. <http://dx.doi.org/10.1109/iccgi.2008.25>
- [18] Brown, P. J., & Jones, G. J. (2001). Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. Personal and Ubiquitous Computing, 5(4), 253-263. <http://dx.doi.org/10.1007/s007790170004>
- [19] José, R., Moreira, A., Rodrigues, H., & Davies, N. (2003). The AROUND architecture for dynamic location-based services. Mobile Networks and Applications, 8(4), 377-387. <http://dx.doi.org/10.1023/A:1024531629631>

#### AUTHORS

**Yousef Ibrahim Daradkeh** is with the College of Engineering at Wadi Aldawaser, Prince Sattam bin Abdulaziz University, KSA ([daradkeh@yahoo.ca](mailto:daradkeh@yahoo.ca))

**Dmitry Namiot** is with the Faculty of Computational Math and Cybernetics, Lomonosov Moscow State University, Moscow, Russia ([dnamiot@gmail.com](mailto:dnamiot@gmail.com)).

This work was supported in part by the Deanship of Scientific Research, Prince Sattam bin Abdulaziz University, Saudi Arabia. Project number is 2014/1/863. Submitted 24 April 2015. Published as resubmitted by the authors 20 August 2015.